

Logic

Exercise Sheet 6

Exercise 6.1 Validity and satisfiability

Which of the following formulas is valid, which is satisfiable?

For each satisfiable formula give a model. For each formula that is satisfiable but not valid, give a counter-model.

- a) $(\forall x.\exists y.P(x,y)) \Rightarrow (\exists y.\forall x.P(x,y))$
- b) $(\exists x.\forall y.P(x,y)) \Rightarrow (\forall y.\exists x.P(x,y))$
- c) $(\exists x.\forall y.P(x,y)) \Rightarrow (\forall x.\exists y.P(x,y))$

Solution

It follows from Exercise 6.3 that all three formulas are satisfied by the trivial one-element model M with $D_M = \{*\}$ and $P_M(*, *) = \text{true}$.

a) is not valid. As a countermodel, take for example $D_M = \mathbb{N}$ and $P_M(n, m) = n < m$.

b) is valid. *Proof:*

Let M be an arbitrary interpretation. If $M \models \exists x.\forall y.P(x, y)$ then there exists an $a \in D_M$ such that for all $b \in D_M$, $P_M(a, b)$ holds. hence in particular $M \models \forall y.\exists x.P(x, y)$.

Remark: This proof is a little unsatisfying since all we did is interpret the formula and then do some informal reasoning on semantic level. This already shows that we will need a calculus, a set of rules to form proofs. We will later use resolution and the sequent calculus for such proofs.

c) is not valid. A countermodel is $D_M = \{\diamond, \square\}$ and $P_M(\diamond, y) = \text{true}$ and $P_M(\square, y) = \text{false}$.

Exercise 6.2 More validity, satisfiability and models

Prove or disprove, using the definitions:

- a) For each sentence p , either p is valid or $\neg p$ is valid.
- b) A sentence p is valid iff $\neg p$ is unsatisfiable.

Solution

- a) The statement is false. Take for example the sentence $p = \forall x.P(x)$. Both p and $\neg p$ are invalid. A simple countermodel for p is $D_M = \mathbb{N}$ where $P_M(n) = \text{true}$ iff n is even. A countermodel of $\neg p$ is the trivial model from Exercise 6.3.

b) *Detailed Proof:*

\Rightarrow : Assume that p is valid. We show that $\neg p$ is unsatisfiable. Let M be an arbitrary interpretation. Then $M \models p$ since p is valid. For an arbitrary valuation v we have $\text{holds } M v (\neg p) = \text{not } (\text{holds } M v p) = \text{false}$. So $M \not\models \neg p$. Since M was chosen arbitrarily, it follows that $\neg p$ does not have a model.

The reverse direction follows the same pattern.

Exercise 6.3 Trivial model

Prove that a formula containing only $\wedge, \vee, \forall, \exists, \Rightarrow$ and atomic formulas is always satisfiable.

Solution

We use the singleton model with $D_M = \{*\}$ and $P_M(*, \dots, *) = \text{true}$ for each predicate symbol P . For function symbols there is only one possible interpretation anyway: $f_M(*, \dots, *) = *$.

Now it is a straightforward induction to show that for any valuation v , $\text{holds } M v p$:

Case $p = \top$: obvious.

Case $p = P(t_1, \dots, t_n)$: Note that $\text{termval } M v t_i = *$ for all i . Hence

$$\text{holds } M v (P(t_1, \dots, t_n)) = P_M(*, \dots, *) = \text{true} .$$

Case $p = p_1 \wedge p_2$: Then

$$\begin{aligned} & \text{holds } M v (p_1 \wedge p_2) \\ &= (\text{holds } M v p_1) \text{ and } (\text{holds } M v p_2) \\ &= \text{true and true} && \text{(by ind.hyp.)} \\ &= \text{true} \end{aligned}$$

The cases for \vee and \Rightarrow are similar. We consider the quantifiers:

Case $p = \forall x.p'$: Then

$$\begin{aligned} & \text{holds } M v (\forall x.p') \\ &= \text{for all } a \in D_M, \text{ holds } M ((x \mapsto a)v) p' \\ &= \text{for all } a \in D_M, \text{ true} && \text{(by ind.hyp.)} \\ &= \text{true} \end{aligned}$$

The case for \exists is similar.

We conclude that $M \models p$. Of course this would not work if the formula contained \neg or \perp .

Remark: Note that we did not use the fact that the model has only one element. In fact our proof shows that any interpretation where predicates are always true is a model of p .

Exercise 6.4 Extension with $\exists!$

- a) Extend the definition of first-order formulas with an additional quantifier $\exists!$, such that $\exists!x.P(x)$ means that there exists a unique x such that $P(x)$ holds. Modify the definition of the semantics.
- b) Extend the OCaml implementations of the type `formula` and the function `holds` accordingly.
- c) Is the following formula valid?

$$(\exists!x.\exists!y.P(x, y)) \Rightarrow (\exists!y.\exists!x.P(x, y))$$

Solution

- a) The extension of the syntax is already given. For the semantics, we extend the definition of `holds` with the clause

$$\begin{aligned} \text{holds } M \ v \ (\exists!x.p) &= \text{for some } a \in D_M \text{ and all } b \in D_M \setminus \{a\}, \\ &\text{holds } M \ ((x \mapsto a)v) \ p \text{ and not } \text{holds } M \ ((x \mapsto b)v) \ p \end{aligned}$$

- b) (omitted)
- c) The formula is not valid, as the following countermodel shows: Let $D_M = \{a, b, c\}$. The interpretation for P is given by the following truth table, where \times marks a true entry:

$P_M(x, y)$	$y = a$	b	c
$x = a$	\times		
b	\times	\times	\times
c			

Remark: Adding $\exists!$ to the language is not just a trivial abbreviation but has interesting consequences: For example, we can show that any model M of the formula $(\forall x.P(x)) \wedge (\exists!x.P(x))$ must satisfy $|D_M| = 1$. In Exercise 7.1. we show that this is not possible with just \forall and \exists . Moreover, we can even axiomatize equality by the formula $\forall x.Eq(x, x) \wedge \exists!y.Eq(x, y)$.